

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE 15 Jun 94	3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE Test and Evaluation Software Operational Assessment Guide			5. FUNDING NUMBERS
6. AUTHOR(S) Capt. Delmar Gassner			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Operational Test and Evaluation Center Directorate of Systems Analysis 8500 Gibson Blvd. S.E. Kirtland AFB NM 87117-5558			8. PERFORMING ORGANIZATION REPORT NUMBER AFOTEC Pamphlet 99-102 Volume 8
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
			
11. SUPPLEMENTARY NOTES AFOTEC Pamphlet 99-102 replaces AFOTEC Pamphlet 800-2, all volumes.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified/Unlimited		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) This pamphlet describes the method and procedures used by the Air Force Operational Test and Evaluation Center (AFOTEC) for performing software operational assessments (SOA) for both embedded computer systems and traditional stand-alone computer systems.			
14. SUBJECT TERMS			15. NUMBER OF PAGES 36
16. PRICE CODE			
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited

## GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to *stay within the lines* to meet *optical scanning requirements*.

### **Block 1. Agency Use Only (Leave blank).**

**Block 2. Report Date.** Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

### **Block 3. Type of Report and Dates Covered.**

State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

**Block 4. Title and Subtitle.** A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

**Block 5. Funding Numbers.** To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

<b>C</b> - Contract	<b>PR</b> - Project
<b>G</b> - Grant	<b>TA</b> - Task
<b>PE</b> - Program Element	<b>WU</b> - Work Unit
	Accession No.

**Block 6. Author(s).** Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

**Block 7. Performing Organization Name(s) and Address(es).** Self-explanatory.

**Block 8. Performing Organization Report Number.** Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

**Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es).** Self-explanatory.

**Block 10. Sponsoring/Monitoring Agency Report Number. (If known)**

**Block 11. Supplementary Notes.** Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

### **Block 12a. Distribution/Availability Statement.**

Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

**DOD** - See DoDD 5230.24, "Distribution Statements on Technical Documents."

**DOE** - See authorities.

**NASA** - See Handbook NHB 2200.2.

**NTIS** - Leave blank.

### **Block 12b. Distribution Code.**

**DOD** - Leave blank.

**DOE** - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.

**NASA** - Leave blank.

**NTIS** - Leave blank.

**Block 13. Abstract.** Include a brief (*Maximum 200 words*) factual summary of the most significant information contained in the report.

**Block 14. Subject Terms.** Keywords or phrases identifying major subjects in the report.

**Block 15. Number of Pages.** Enter the total number of pages.

**Block 16. Price Code.** Enter appropriate price code (*NTIS only*).

**Blocks 17. - 19. Security Classifications.** Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

**Block 20. Limitation of Abstract.** This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.



**Test and Evaluation**

**SOFTWARE OPERATIONAL ASSESSMENT GUIDE**

This pamphlet describes the method and procedures used by the Air Force Operational Test and Evaluation Center (AFOTEC) for performing software operational assessments (SOA) for both embedded computer systems and traditional stand-alone computer systems.

AFOTEC Pamphlet 99-102 replaces AFOTEC Pamphlet 800-2, all volumes. This volume is one in a series of software operational test and evaluation (OT&E) guidelines prepared by the Software Analysis Team of the AFOTEC Systems Analysis Directorate. It is intended for use in operational test and evaluation of software. Local reproduction of all volumes in this series is authorized. Direct any comments to the office of primary responsibility (OPR). The series is as follows:

**AFOTEC Pamphlet 99-102, Volume 1 - Management of Software Operational Test and Evaluation**

**AFOTEC Pamphlet 99-102, Volume 2 - Software Support Life Cycle Process Evaluation Guide**

**AFOTEC Pamphlet 99-102, Volume 3 - Software Maintainability Evaluation Guide**

**AFOTEC Pamphlet 99-102, Volume 4 - Software Usability Evaluation Guide**

**AFOTEC Pamphlet 99-102, Volume 5 - Software Support Resources Evaluation Guide**

**AFOTEC Pamphlet 99-102, Volume 6 - Software Maturity Assessment Guide**

**AFOTEC Pamphlet 99-102, Volume 7 - Software Reliability Evaluation Guide**

**AFOTEC Pamphlet 99-102, Volume 8 - Software Operational Assessment Guide**

**1. Purpose.** SOAs are an integral part of a system's operational assessment (OA). The central theme of an OA is interaction with the operating command and developer to:

1.1. Provide a perspective on a system's progress toward operational effectiveness and suitability and to determine if there are deficiencies or voids that have a major impact on operational effectiveness and suitability. Using development test data, we will be able to assess how the system is progressing toward meeting user requirements. Or:

1.2. Assess a system's readiness for OT&E.

To achieve this goal, this guide describes techniques for the AFOTEC software test manager (STM) and deputy for software evaluation (DSE) to use in assessing aspects of computer resources and software during a system OA. This guide provides the necessary information to plan, execute, and report the SOA regardless of the system's development phase.

**2. Document Organization and Use.**

2.1. The following pages describe how to use this guide in planning, executing, and reporting the SOA and provide:

2.1.1. A background of the AFOTEC SOA methodology.

2.1.2. A basic understanding of the assessment procedures.

2.1.3. Attachment 1, which describes the assessment factors and subfactors for use in any assessment area.

2.2. This guide will help you:

2.2.1. Define the SOA objectives.

2.2.2. Select the factors relevant to the SOA.

2.2.3. Collect and analyze background and supporting data.

2.2.4. Assign ratings to the factors.

2.2.5. Analyze and interpret the results to determine the rating for each assessment area.

2.2.6. Report the results.

Prior to beginning the SOA, review AFOTEC Pamphlet (AFOTECPAM) 99-102, volume 1 to gain an understanding of AFOTEC's software OT&E process.

**3. Operational Assessment Areas.** There are five standard areas to assess during the OA of a target program. The SOA assessment areas mirror the OA assessment areas:

- Area 1. Major impacts affecting operational effectiveness and suitability.
- Area 2. Programmatic voids.
- Area 3. Adequacy of requirements.
- Area 4. Ability to support adequate operational testing.
- Area 5. Results from higher headquarters-directed system assessment activities.

Area 1, where we examine developmental or other relevant test data, is required for all SOAs, with the possible exception of higher headquarters-directed specific assessments

(area 5). The assessment areas are introduced in the following subsections (see AFOTEC Instruction (AFOTECI) 99-101 for further clarification).

### **3.1. Area 1 (Major Impacts).**

**3.1.1. Scope.** Identify and assess major impacts affecting operational effectiveness and suitability.

#### **3.1.2. Assess:**

3.1.2.1. Areas of risk (known areas of risk in developing the system).

3.1.2.2. Results of any software testing to identify significant trends noted in development efforts.

### **3.2. Area 2 (Programmatic Voids).**

**3.2.1. Scope.** Identify any software or computer resource programmatic voids that would adversely impact the ability of the system to meet operational requirements.

**3.2.2. Assess.** The program information/documentation to determine if software development/testing progresses as planned, the system will meet operational requirements.

### **3.3. Area 3 (Adequacy of Requirements).**

**3.3.1. Scope.** Review the status of software documentation, with emphasis on user requirements development including completeness, clarity, sufficiency, priority, rationale, or other factors that could affect software testability.

**3.3.2. Assess.** The software-related documents to determine if the required software and computer resource information is contained therein. The absence or inadequacy of critical documents will seriously impact the program.

### **3.4. Area 4 (Ability to Support Adequate Operational Testing).**

**3.4.1. Scope.** Review the schedule for the OT&E events.

**3.4.2. Assess:**

3.4.2.1. Software schedule to accommodate OT&E activities.

3.4.2.2. Software development/major problems that could prohibit conduct of OT&E on a properly configured test article.

3.4.2.3. Required software and computer resources to support OT&E.

3.4.2.4. Software development and maturity aspects that would impact OT&E.

**3.5. Area 5 (Higher Headquarters Results).**

3.5.1. **Scope.** Review/observe special field test activities or other efforts when specifically requested.

3.5.2. **Assess.** Higher headquarters-directed software and computer resource issues.

**4. Planning.** The following considerations are necessary to plan an effective SOA:

**4.1. Select Relevant Factors.** You must clearly define the procedures for performing the SOA. To do this, identify the factors, tools, and program data resources relevant to a specific program assessment. The SOA team should begin by reviewing attachment 1 to select the factors relevant to the assessment of each of the areas. Eliminate factors or subfactors that are unnecessary or premature for the current program stage of development. Also review the statement of work (SOW) and contract data requirements lists (CDRL) to understand the required design, development, testing, and quality processes.

**4.2. SOA Resources and Other Considerations.** There will be constraints on the SOA team in terms of time, funds, and personnel. Consider the following when developing the SOA plan:

**4.2.1. Personnel Resources.** In most cases, the STM or DSE, using AFOTEC resources, will accomplish the SOA. The combination of personnel experience, time allowed, and pro

gram size will determine the breadth and depth of the SOA.

**4.2.2. Program Data Resources.**

4.2.2.1. Review program status information to include government and deliverable documentation, program reviews, and interviews.

4.2.2.2. Team members should attend any program reviews prior to, or falling within the time frame of, the SOA, and use key program personnel as information sources.

4.2.2.3. If contractor internal data resources (such as software development files (SDF)) are included in the program resources, arrange with the program office in advance to gain access.

4.2.2.4. Previous evaluations/assessments may provide insight into problem areas or program voids. The SOA team must review any other studies performed, such as previous SOAs, development test and evaluation (DT&E), IV&V or Software Engineering Institute (SEI) assessments to gain information useful to the SOA effort.

**4.2.3. Other Considerations:**

4.2.3.1. Identify travel and funding requirements (for data collection, conducting interviews, attending reviews and computer support) in the test resource plan (TRP).

4.2.3.2. Contact any necessary supporting agencies.

4.2.3.3. Allow for special security considerations.

**5. SOA Execution.** Responses to the factors and subfactors in attachment 1 are the core of the SOA. Use other tools and information sources to provide background information and determine if problem areas are receiving adequate attention.

**5.1. SOA Rating Scale.**

**5.1.1. Green.** No issues or issues receiving adequate attention.

5.1.2. **Yellow.** Issues requiring additional attention.

5.1.3. **Red.** Significant areas of concern or voids exist in schedule, resources, system design, etc., that will impact OT&E and/or may prevent the achievement of specified operational requirements.

5.1.4. **White.** Area not assessed.

**5.2. Recording Responses.** Your responses will depend on your experience and subjective judgment. If you score a factor or subfactor yellow or red, record the impact of such a rating (major, minor, none) and the status of any plan to correct the deficiency (plan in-place, plan in-place but inadequate, no plan). Finally, you must record the rationale for a low rating for a factor or subfactor. Also record supplementary information to provide historical information.

**5.3. Ratings.** Assign a color rating to each subfactor. Combine the subfactor ratings into a single color rating for the parent factor and then combine the factor ratings into a single color for the parent assessment area. Again, use your best judgment coupled with comments and other information recorded for each factor to make a decision as to the overall color rating for an assessment area.

## 6. SOA Inputs to the OA Plan/Report.

6.1. The customer for all SOA inputs and results is the program's operational test manager. The test manager will determine the format for each plan and report. The SOA team will be responsible for providing wording on the following:

6.1.1. Software and computer resource description.

6.1.2. How the software SOA fits into the system OA.

6.1.3. SOA dates and location.

6.1.4. The data sources that were selected to address the software and the computer

resource measures of performance (MOP) and measures of effectiveness (MOE).

6.1.5. The user requirements for the software and computer resource MOPs and MOEs.

6.1.6. How test data results can be utilized to examine software and computer resource MOPs and MOEs under a specific COI and draw effectiveness and suitability conclusions.

6.1.7. Anticipated planning considerations and limitations that may affect the SOA concept or scope.

6.1.8. List those areas and factors to be examined during the SOA. Keep in mind that we include Area 1 in every SOA.

6.1.9. Provide an overview of how the SOA will be or was conducted.

6.1.10. List planning considerations and limitations that shaped the SOA design or affected assessment results.

6.1.11. State overall conclusions with a list of recommendations (if appropriate).

6.2. After conclusion of the SOA, include any lessons-learned in SAS's SOA database.

## 7. Summary of Assessment Philosophy.

The following is a summary of the general philosophy to guide you in assessing the system:

7.1. Your primary consideration: "Are the system's software and computer resource progressing toward operational effectiveness and suitability, and are these functions ready to enter the next phase of the life cycle?"

7.2. Some necessary resources may not be available at the time of the SOA. You must tailor out those areas which do not apply to the system or are not available. However, if items are not available that should be available at that point in the system's life cycle, you should rate them lower.

7.3. Finally, bear in mind you were chosen for this SOA because of your demonstrated

expertise. That expertise and the professionalism you demonstrate in completing this

assessment will help provide the Air Force with a quality software product.

GEORGE B. HARRISON, Major General, USAF  
Commander

1 Attachment

1. Software Operational Assessment (SOA) Factors.

Accesion For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution / _____	
Availability Codes	
Dist	Avail and / or Special
A-1	

**SOFTWARE OPERATIONAL ASSESSMENT (SOA) FACTORS****INTRODUCTION**

This attachment contains a list of factors and subfactors you can use in any of the five SOA assessment areas. AFOTECPAM 99-102, volume 2, attachment 3 contains a list of the Software Support Life Cycle Process questions that should be answerable at the various time phases or milestone reviews. The factors are grouped in five categories (NOTE: The categories are just placeholders to group related factors):

1. Program Schedule Adequacy.
2. Program Resources.
3. Documentation Status.
4. Development/Maturity.
5. Requirements Traceability.

Assess only those factors and subfactors applicable to this SOA. You can determine ratings at the factor level or the subfactor level. The factor and subfactors are statements describing desirable attributes of a program's development and testing processes. When response instructions are not provided, respond to the factors and subfactors based on your experience and expertise. You may tailor the individual factors and subfactors or add factors/subfactors of your choice.

Listed below are the categories, factors, and subfactors available.

**1. Program Schedule Adequacy:****1.1. Development Schedule Adequacy.****1.2. OT&E Schedule Adequacy.****2. Program Resources:****2.1. Test Article Availability.****2.2. Test Resources.****2.3. CRWG.****3. Documentation Status:****3.1. Planning Documents:****3.1.1. TEMP.****3.1.2. OT&E Plan.****3.1.3. TRP.****3.1.4. CRLCMP.**

**3.2. Requirements Documents:**

**3.2.1. ORD.**

**3.2.2. RCM.**

**4. Development/Maturity:**

**4.1. Software Development Processes:**

**4.1.1. Audits/Reviews.**

**4.1.2. Configuration Management:**

**4.1.2.1. Identification.**

**4.1.2.2. Control.**

**4.1.2.3. Audit.**

**4.1.2.4. Accounting.**

**4.1.3. Quality Assurance.**

**4.1.4. Program Management:**

**4.1.4.1. Resource Functions.**

**4.1.4.1.1. Planning.**

**4.1.4.1.2. Organization.**

**4.1.4.1.3. Monitoring.**

**4.1.4.2. Control Functions.**

**4.1.4.2.1. Corporate Policies.**

**4.1.4.2.2. Subcontractor Management.**

**4.1.4.2.3. Communication Interfaces.**

**4.1.5. Testing.**

**4.2. Products:**

**4.2.1. Documents.**

**4.2.1.1. SDP.**

**4.2.1.2. SSDD.**

**4.2.1.3. IDD.**

**4.2.1.4. SQPP.**

4.2.1.5. SCMP.

4.2.1.6. SDD.

4.2.1.7. STP.

4.2.1.8. SRS.

4.2.1.9. IRS.

4.2.2. Software.

4.2.2.1. Reliability.

4.2.2.2. Schedules.

4.2.2.3. Spare Capacity.

4.2.2.4. Maintainability.

5. Requirements Traceability:

5.1. ORD/RCM to System Specification Traceability.

5.2. Development Contractor Traceability.

## 1. PROGRAM SCHEDULE ADEQUACY

The program's master schedule must clearly show schedules adequate to support computer resource DT&E and OT&E. The master schedule must incorporate current status information of the software development schedule. You need to know what is happening on a broader scale with software development/testing and how identified risk areas might impact the ability to meet program schedules.

**FACTOR:** The program's master schedule is adequate to support computer resource development, DT&E, and OT&E.

### SUBFACTORS:

DEVELOPMENT SCHEDULE ADEQUACY (1.1) - The software development schedule is adequate.

OT&E SCHEDULE ADEQUACY (1.2) - The OT&E schedule is adequate to support computer resources and software evaluations.

**EXPLANATION:** Many activities must be identified and shown on the schedule to ensure they are tracked for proper completion.

**REFERENCE QUESTIONS (NOTE - see AFOTECPAM 99-102, volume 2, Software Life Cycle Process Evaluation Guide for expansion of the reference questions):**

SPM(PL)-3 Planning for computer resources has been based on an acquisition schedule with adequately specified milestones.

SPM(PL)-16 Planning for DT&E of computer resources has been adequate throughout the system life cycle.

- SPM(OS)-1 The software requirements have been adequately allocated to elements of work breakdown structure (WBS).
- SPM(OS)-2 The software related tasks are clearly identified in the WBS.
- SPM(OS)-3 The key project personnel and their assignments in relation to the WBS software-related tasks are clearly identified.
- SPM(OS)-4 The coordination of modifications to the WBS among all activities has been adequate.
- SPM(TS)-6 The requirements for the development contractor software test strategy are clearly specified in the RFP, SOW, and/or CDRLs.

### 1.1. DEVELOPMENT SCHEDULE ADEQUACY

**FACTOR:** The software development schedule is adequate.

**EXPLANATION:** It is critical the software development work meets the overall schedule. Specifically, investigate the progress on software-related tasks, but also identify any preceding tasks upon which the software development activity depends. The Software Development Plan will document the software development schedule.

- The WBS should include computer resource activities such as major reviews and milestones and be used as the basis for schedule and cost estimates. The WBS should be updated throughout the program to reflect significant changes.
- The critical path of the schedule should be identified.
- Confirm each computer resource schedule milestone is a measurable event.
- The SPO should have developed and exercised a software risk abatement plan.
- The program schedule must also include adequate time for DT&E. In addition to time to execute DT&E, the program schedule should include sufficient time for any system rework that is required based on DT&E findings.

**RESPONSE INSTRUCTIONS:** The evaluator should look for impacts which have not been reflected in the *current* schedule and, therefore, make it unrealistic or unachievable. For the current schedule, if all significant major milestones or key software deliverable dates are slipping, or have slipped, or show signs of converging to unrealistic dates, the evaluator should consider assigning a red to this factor.

**RELATED METRICS:** Examine a schedule metric that indicates adherence to the planned schedules for major software milestones and key software deliverables. Compute the planned and actual schedules for major milestones and key software deliverables as they change over time. (See AFP 800-48.)

### 1.2. OT&E SCHEDULE ADEQUACY

**FACTOR:** The OT&E schedule is adequate to support the computer resources and software evaluations.

**EXPLANATION:** Initial information on activities to be performed during the OT&E may be obtained from the test concept and the TEMP. The test concept should identify an initial OT&E schedule. The OT&E plan should contain test objectives and evaluation methods to assess the software. The schedule should include activities for both system effectiveness (i.e., system performance objectives) and system suitability evaluations (i.e., maintainability). Obtain the adequacy of the schedule to assess system effectiveness and suitability by examining test mission scenarios and the Data Man-

agement and Analysis Plan (DMAP) for tests that include software components. Assess the software aspects of these tests and cross-reference them to the functional software requirements of the system.

In addition to time allocated for OT&E, there should be sufficient slack time so any slippage of preceding activities (such as DT&E) does not impact OT&E.

#### REFERENCE QUESTION:

SPM(PL)-17 Planning for OT&E of computer resources has been adequate throughout the system life cycle.

## **2. PROGRAM RESOURCES**

Resources required by the program must be addressed in the program's planning process. The plan should account for the availability of all components such as personnel, equipment, and facilities required to meet the schedule and describe the special characteristics of software development. Planning should identify specific test articles and test resources and ensure their availability. In addition, the presence of the CRWG, which coordinates many of the processes associated with software issues, must be addressed.

**FACTOR:** The planned computer resources are adequate to support software development, DT&E, and OT&E.

#### SUBFACTORS:

TEST ARTICLE AVAILABILITY (2.1) - The required test articles are planned to be available for DT&E and OT&E of computer resources.

TEST RESOURCES (2.2) - The test computer resources required to support the software testing objectives are adequately planned and will be available for DT&E and OT&E.

CRWG (2.3) - The CRWG is operational.

**EXPLANATION:** The successful completion of DT&E and OT&E requires the availability of identified and controlled software test articles. In addition, planning must occur for the availability of hardware and software as well as evaluation tools and personnel to perform the evaluations.

### **2.1. TEST ARTICLE AVAILABILITY**

**FACTOR:** The required test articles will be available for DT&E and OT&E of computer resources.

**EXPLANATION:** The primary sources for information on what software products should be provided are the program Statement of Work (SOW), the program CDRL, and the OT&E plan. A list of test article items, or "first article test items," should be provided in these documents. Once they are identified, you should compare the software test articles with a schedule for their delivery and ensure the schedules for software development, DT&E, and OT&E indicate the software test articles will be available by the need dates.

For OT&E to be successfully completed, the software products must be complete, certified, and provided in the quantities required. Software should be patch-free and certified by the developing agency prior to use during the OT&E. All software products provided for OT&E should be under strict software configuration control by the contractor. Plans for these activities should be documented in the SOW and the SDP.

Included in the definition of software test articles are all deliverables (see the CDRLs) required for the system, to include software documents required to operate and maintain the system. These articles will be subjected to supportability evaluations during the OT&E. Reference the development/maturity category for details on determining the system development/maturity issues.

## 2.2. TEST RESOURCES

**FACTOR:** The test computer resources required to support the software testing objectives are adequately planned and will be available for DT&E and OT&E.

**EXPLANATION:** Plans must exist describing the hardware on which the software to be tested resides. In addition to the software under development, any additional support software and databases required for testing need to be available, populated with the required data, and validated. Capabilities and limitations of all software tools (and hardware resources, interfaces, etc.) need to be documented and available to the test team.

Use the test concept, TRP, the CDRL, the OT&E plan, and the SDP to identify test resources. The OT&E plan should include specific evaluation methods for software. This plan should specify what type of evaluations are planned and what resources are required for each one. Other sources of information may be agreements between AFOTEC, the SPO, and using commands.

### **REFERENCE QUESTION:**

SPM(TS)-18 The software test environment (host emulations/simulation, integrated laboratory environments, software bench testing equipment, and operational mission test environments) is adequately identified in the software test documentation and is adequate for accomplishing the required testing.

## 2.3. CRWG

**FACTOR:** The CRWG is operational.

**EXPLANATION:** The CRWG should be established prior to Milestone I. Its function is to provide guidance on computer resources issues throughout the development effort. The CRWG is responsible for making key recommendations to the program manager and for describing the support concept in the CRLCMP. It should include members from the operating, supporting, and participating commands.

The CRWG participates in program management reviews, source selection evaluation boards, solicitation review, design reviews, and audits. Actions of the CRWG may help in identifying activities that should occur for the system to be ready for OT&E. The CRWG also supports the TPWG as required. The CRWG should exist, and it should be active. CRWG meeting minutes should be available for examination.

**RESPONSE INSTRUCTIONS:** If a CRWG has not been established, or is not meeting on a periodic basis, the evaluator should consider rating this area as red.

### **REFERENCE QUESTIONS:**

SPM(PL)-12 The CRWG organization has been adequate throughout the system life cycle.

SPM(PL)-13 The CRWG has had clearly specified responsibilities and appropriate authority to implement those responsibilities throughout the system life cycle.

SPM(PL)-14 The CRWG has properly ensured that computer resources comply with established policy, procedures, plans and standards.

SPM(PI)-9 The CRWG external interfaces are adequate.

### 3. DOCUMENTATION STATUS

The objective of the documentation status factors is to review the critical documents (TEMP, OT&E plan, TRP, and CRLCMP) to determine whether they have progressed sufficiently to support the DT&E and OT&E. In addition, the ORD and RCM are reviewed to determine the adequacy of the requirements, specifications, and test criteria (e.g., prioritized, operationally stated, no disconnects).

**FACTOR:** The documentation is progressing adequately to support DT&E and OT&E of computer resources.

#### SUBFACTORS:

PLANNING DOCUMENTS (3.1) - The planning documents are progressing adequately to support OT&E of computer resources.

REQUIREMENTS DOCUMENTS (3.2) - The requirements documents are progressing adequately to support OT&E of computer resources.

#### 3.1. PLANNING DOCUMENTS

**FACTOR:** The planning documents are progressing adequately to support DT&E and OT&E of computer resources.

#### SUBFACTORS:

TEMP (3.1.1) - The TEMP contains the appropriate computer resource information.

OT&E PLAN (3.1.2) - The OT&E plan contains the appropriate computer resource information.

TRP (3.1.3) - The TRP includes the appropriate computer resource information.

CRLCMP (3.1.4) - The CRLCMP contains the appropriate computer resource information.

**EXPLANATION:** Factors have been included for the following planning documents: TEMP, OT&E plan, TRP, and CRLCMP. Additional (planning) documents that can be reviewed include the reliability and maintainability management plan, MOA/MOU, technical orders, and the Joint Reliability and Maintainability Evaluation Team (JRMET) charter.

The documents should be updated when there are significant changes to the program, at milestones, and at other major events. There should be a process in place to ensure all documents are updated to reflect changes in the requirements.

##### 3.1.1. TEMP

**FACTOR:** The TEMP contains the appropriate computer resources information.

**EXPLANATION:** The TEMP should summarize the critical computer resource operational effectiveness and suitability parameters. There should be no disconnects in computer resource requirements between the ORD, RCM, and the TEMP.

The TEMP should list the critical software technical parameters of the system with the accompanying objectives and thresholds. These critical technical parameters are derived from the ORD, critical system characteristics, and technical performance measures. The TEMP should also address support concepts (i.e., postdeployment software support) resulting in special test and analysis requirements.

The TEMP should address computer resources testing to include the level of stress testing that will be required (e.g., out of range, load, alternative sign, maximum and minimum value).

#### REFERENCE QUESTIONS:

- SPM(PL)-1 Planning for computer resources has been adequate with respect to acquisition development, logistics, and training.
- SPM(TS)-3 The software test process for OT&E has followed the guidelines in the TEMP.

#### **3.1.2. OT&E PLAN**

**FACTOR:** The OT&E plan contains the appropriate computer resource information.

**EXPLANATION:** The OT&E plan should contain the appropriate computer resources evaluation criteria. These criteria are tied directly to the operating command's stated computer resource operational requirements and represent a level of performance against which system characteristics are compared.

The computer resource objectives that are addressed in the OT&E plan should be consistent with the computer resources information in other documents that address this type of information (specifically the TRP and TEMP).

The OT&E plan should spell out the software test readiness criteria. AFOTEC/SAS has developed a boilerplate of typical software OT&E test readiness requirements for a program that is certified ready to begin OT&E. If the OT&E plan is not available, review the test concept or Test Process Review (TPR) notes.

**RESPONSE INSTRUCTIONS:** Cross-check the OT&E evaluation criteria with those in the RCM and with the ORD requirements to ensure completeness, applicability, and reasonableness. Any disagreements between the documents should be noted and the factor rated yellow or red depending upon the severity of the disconnect.

#### **3.1.3. TRP**

**FACTOR:** The TRP includes the appropriate computer resource information.

**EXPLANATION:** The TRP should identify requirements for computer resources to include hardware, software, supplies, and personnel (e.g., evaluators, DSE).

All software evaluation personnel must be identified in the TRP. This identification must be specified for each fiscal year and include the title, grade, number, source (i.e., AFMC), number of days per person, dates required, site, and type of assignment (e.g., TDY, PCS). The training for the DSE and software evaluator should also be identified in the TRP.

#### **3.1.4. CRLCMP**

**FACTOR:** The CRLCMP contains the appropriate computer resource information required to support DT&E and OT&E.

**EXPLANATION:** The CRLCMP should describe the functional interrelationships among the following organizations: implementing command, using command, participating commands (testing, training, and contract administration), and other agencies. These descriptions should include an identification of the major functions and responsibilities that relate to computer resources. CRLCMP requirements are listed in AFR 800-14.

**RESPONSE INSTRUCTIONS:** If any of the AFR 800-14 information is missing from the CRLCMP, the risk of problems during testing is higher. Use judgment in determining if missing information is sufficient reason to rate this factor yellow or red.

**REFERENCE QUESTIONS:**

- SPM(PL)-9 The CRLCMP contains adequate specifications of the acquisition requirements for computer resources.
- SPM(PL)-10 The CRLCMP adequately addresses the responsibilities and procedures to ensure proper software configuration management throughout the system life cycle.

**3.2. REQUIREMENTS DOCUMENTS**

**FACTOR:** The requirements documents are progressing adequately to support DT&E and OT&E of computer resources.

**SUBFACTORS:**

- ORD (3.2.1) - The ORD has reached a level of completion consistent with the schedule.
- RCM (3.2.2) - The RCM contains the appropriate computer resource information.

**EXPLANATION:** The requirements documents are the ORD and the RCM. These documents should contain the computer resources requirements. These requirements should be stated in operational terms and be testable in order to support DT&E and OT&E. (See AFI 1-601 for ORD/RCM requirements.)

**3.2.1. ORD**

**FACTOR:** The ORD has reached a level of completion consistent with the schedule.

**EXPLANATION:** The ORD contains the operating command's documentation of essential quantitative and qualitative requirements and describes how the system will be used. As requirements change throughout the system development process, the ORD should be updated to reflect the changes.

The critical computer resource requirements described in the ORD appear as parameters in the RCM. The parameters in the RCM should be predicated upon requirements in the ORD. You should identify the critical computer resource requirements in the ORD and ensure they are in the RCM. Use the following template for requirements that should be in the ORD:

- (1) Requirements identifying computer resource constraints (e.g., language, computer).
- (2) Requirements addressing all ILS elements for computer resources.
- (3) Requirements addressing plans for using command support of software.
- (4) Requirements for software that support user-friendly operations and maintenance.
- (5) Software maturity requirements.
- (6) A requirement for when the software support agency(s) must be functional.
- (7) Requirements for high quality software source code, documentation, and architecture design.
- (8) Environmental computer resource constraint
- (9) Requirements for timing.
- (10) Requirements for spare memory and throughput.
- (11) Requirements for computer memory and throughput growth.
- (12) A requirement for development of a software support facility.
- (13) A requirement for software configuration control.

- (14) Accuracy requirements for each applicable function.
- (15) Error handling requirements.
- (16) Interface capability requirements.

### 3.2.2. RCM

**FACTOR:** The RCM contains the appropriate computer resource information.

**EXPLANATION:** The RCM documents and tracks creation of and changes to user requirements. Its purpose is to provide a comparison and correlation of requirements to specifications and test criteria. It is a mandatory attachment to the ORD. The RCM contains a comparison of the user's system requirements, contractual specifications, and operational evaluation criteria.

An overabundance of changes to the RCM may indicate instability in the requirements. The RCM shows current requirements or goals from the most recent ORD side-by-side with those from any previous ORD to provide an audit trail of requirements evaluation.

The parameters in the RCM must be stated in quantifiable terms so requirements, specifications, and test criteria can be applied to them.

Computer resource requirements must not be contradictory (i.e., a requirement for all software to be written in a high order language and another requirement for flight control software to be written in assembly language).

The computer resource requirements should be operationally oriented as opposed to specification oriented.

The computer resource requirements should be testable so test cases could reasonably be set up.

#### **REFERENCE QUESTIONS:**

- SPM(PL)-25 The procurement and operational support planning documents have been adequately updated as living documents throughout the system life cycle.
- SPM(DM)-9 The profile of changes to software requirements is reasonable.

### 4. DEVELOPMENT/MATURITY

Maturity is defined in AFOTECPAM 99-102, volume 6, as a "measure [of] the software's progress in its evolution toward satisfying documented user requirements." The objective of these factors is to determine if any problems exist in the software development processes or products that will impact DT&E and OT&E. To meet this objective, you must assess the software processes and the software product in terms of development/maturity.

Problems in the software development processes are identified by assessing the audit and review, configuration management, quality assurance, program management, and testing processes. Audits and reviews provide insights into the maturity of the software products. Problems in the software development products are identified by assessing the software documentation and source code. A review of the software documents will determine their level of completion and whether they contain the appropriate computer resources information. A review of the source code will reveal whether it is progressing sufficiently.

**FACTOR:** There are no significant development/maturity problems in the software development effort that would impact DT&E and OT&E.

**SUBFACTORS:**

\_\_\_\_\_ SOFTWARE DEVELOPMENT PROCESSES (4.1) - The maturity of the software development processes is adequate to ensure properly configured operational software.

\_\_\_\_\_ SOFTWARE DEVELOPMENT PRODUCTS (4.2) - Development and maturity of the software products are adequate to ensure properly configured operational software.

**4.1. SOFTWARE DEVELOPMENT PROCESSES**

**FACTOR:** The maturity of the software development processes is adequate to ensure properly configured operational software.

**SUBFACTORS:**

\_\_\_\_\_ AUDITS/REVIEWS (4.1.1) - The software development audit and review processes indicate development of a maturing product.

\_\_\_\_\_ CONFIGURATION MANAGEMENT (4.1.2) - The software configuration management (SCM) processes are adequate.

\_\_\_\_\_ QUALITY ASSURANCE (4.1.3) - The software quality assurance policies and procedures are effective.

\_\_\_\_\_ PROGRAM MANAGEMENT (4.1.4) - Program management processes in place to determine implementation methodologies, assess progress, and identify shortfalls are adequate.

\_\_\_\_\_ TESTING (4.1.5) - The testing processes are adequate to ensure software is being tested as scheduled, is sufficiently mature, and is satisfying user requirements.

**EXPLANATION:** These processes institutionalize the software development process. All five of the processes must be in place to ensure properly configured software for OT&E.

**REFERENCE QUESTION:**

\_\_\_\_\_ SPM(IM)-012 The contractor's system software tool environment is adequate.

**4.1.1. AUDITS/REVIEWS**

**FACTOR:** The software development audit and review processes indicate development of a maturing product.

**EXPLANATION:** Software development audits and reviews are held to assess the status of the work in progress and to assess the risk associated with proceeding into the next phase of development. Audits and reviews can be used to indicate the product's maturity by evaluating them to discover: if the scheduled audit or review has taken place, if action items identified during the audit or review have been further addressed/resolved, and if the audit or review has been approved. If an audit or review is scheduled and does not take place, this is an indication the product has not progressed sufficiently in its evolution. Further, if there are unresolved action items or an audit or review has been disapproved, this may be an indication of an immature product.

The audits and reviews you could assess to determine process sufficiency are: System Requirements Review (SRR), System Design Review (SDR), Software Specification Reviews (SSR), PDR, CDR, Test Readiness Reviews (TRR), Functional Configuration Audit (FCA), and Physical Configuration Audit (PCA). Note the focus of the evaluation of these audits and reviews is not to assess their quality but

to determine whether they are taking place as scheduled and are identifying and indicating resolution of problem areas. See MIL-STD-1521B, appendices A through H, for an expansion of what each of the audits and reviews is intended to cover.

**RESPONSE INSTRUCTIONS:** If a scheduled audit or review has not taken place, rate the factor red. In addition, action items that have been raised but never further addressed or reviews or audits that have been disapproved should be considered possible reasons for a yellow or red rating depending upon the severity of the issue.

**REFERENCE QUESTIONS:**

- SPM(PL)-004 Computer resources have been adequately addressed as major considerations at procurement reviews, audits, and management evaluations.
- SPM(DM)-010 The profile of unresolved software review action items is reasonable.

**4.1.2. CONFIGURATION MANAGEMENT**

**FACTOR:** The SCM processes are adequate.

**SUBFACTORS:**

- IDENTIFICATION (4.1.2.1)** - Acceptable procedures are used for identifying software baselines.
- CONTROL (4.1.2.2)** - Acceptable procedures to control changes to the baseline are practiced.
- AUDIT (4.1.2.3)** - Acceptable procedures are used to audit development progress for compliance to SCM requirements.
- ACCOUNTING (4.1.2.4)** - Acceptable procedures for reporting configuration management events are practiced.

**EXPLANATION:** SCM processes should be integrated into the software development effort, and an audit trail should exist to document a change and the rationale for the change. The SCM processes should not unnecessarily burden or slow down the development effort and should be automated to the maximum extent possible.

To assess the SCM process, visit the contractor's SCM organization and discuss any issues with the software configuration manager. If this is not possible, an alternative is the SPO SCM organization.

**REFERENCE QUESTIONS:**

- SPM(IM)-011 The contractor's software configuration management support tool environment is adequate.
- SPM(TS)-019 The configuration management of the software test process is adequate.
- SCM(CC)-014 The contractor's automated support tools for configuration control of baselines and internal development identifications is adequate.

**4.1.2.1. IDENTIFICATION**

**FACTOR:** Acceptable procedures are used for identifying software baselines.

**EXPLANATION:** The procedures need to be structured to accomplish at least two functions. First, they need to be structured in a manner that permits identification of both major and minor revisions and numbers of copies. Second, the procedures must coincide with the development of the configuration status accounting tree.

Two types of documents need to be reviewed to rate this factor. First, the software configuration tree needs to be reviewed for adequacy, completeness, and currency. Second, compare the tree structure and numbering to several software configuration items. Use the results of this comparison to rate the factor.

**RESPONSE INSTRUCTIONS:** If the SCM identification procedures are established but are not being adhered to and a corrective action plan exists, rate yellow or green. If SCM identification procedures do not exist and software module partitions exist, rate red, as formal SCM cannot be maintained.

**REFERENCE QUESTIONS:**

- SCM(ID)-003 The procurement activity identification of the software configuration baselines is adequate.
- SCM(ID)-007 The identifier characteristics for software configuration item names are adequate.
- SCM(ID)-010 Contractor's deliverable configuration items are named to adequately identify multiple versions and variations.
- SCM(ID)-011 Contractor's identification procedures are structured to permit easy addition, deletion, or modification of configured items at any hierarchical level.

**4.1.2.2. CONTROL**

**FACTOR:** Acceptable procedures to control changes to the baseline are practiced.

**EXPLANATION:** SCM control procedures need to remain flexible early in the software development to accommodate changing user requirements and/or changes to the software modules and hardware. This can be established by a review of overall Software Change Request (SCR) processing times. Formal control early in the development will adversely affect the development and schedule. As the modules approach FCA/PCA, the SCM control procedures need to be more formal.

The Configuration Control Board (CCB) should be functioning, have the appropriate agency representatives in attendance, and have sufficient authority to implement the control policies. CCB approval should be required to sanction a baseline. To determine these, review the CCB meeting minutes. Accurate up-to-date status, tracking of changes that is timely and responsive, and positive control of each SCR/ECP are attributes of effective SCM control procedures.

**RESPONSE INSTRUCTIONS:** If the procedures are established and adequate but not totally adhered to and a corrective action plan exists, rate the response yellow or green as appropriate. If neither is true, rate the response red.

**REFERENCE QUESTION:**

- SCM(CC)-011 The contractor's configuration control board has an adequate interface with the procurement activity configuration control board.

**4.1.2.3. AUDIT**

**FACTOR:** Acceptable procedures are used to audit development progress for compliance to SCM requirements.

**EXPLANATION:** The SCM audit procedures should be clearly documented. Configuration identification tools can indicate which elements of the configuration identification have been changed as a confirmation of the incorporated changes. Automated tools greatly facilitate the efficiency and accuracy of the audit/review activity.

**RESPONSE INSTRUCTIONS:** If a baseline is established without an audit, or audit procedures are not being followed, rate red.

**REFERENCE QUESTIONS:**

- SCM(AR)-005 The software product acceptance requirements are adequate.
- SCM(AR)-006 The contractor's internal configuration audit/review process facilitates the development of high quality production software.
- SCM(AR)-008 The contractor's configuration management tool support facilitates the audit/review of the process by which changes are incorporated into configuration identifications.

**4.1.2.4. ACCOUNTING**

**FACTOR:** Acceptable procedures for reporting configuration management events are practiced.

**EXPLANATION:** Configuration management accounting deals with the extent and quality to which SCM recordkeeping is performed. SCM procedures to record and report sanctioned baselines, ECPs, SCRs, and software trouble reports need to be established and used at the beginning of the software development. ECP and SCR records from change inception to completion should be complete to provide an audit trail. Software trouble reports should be associated with the software module and track all actions taken.

**RESPONSE INSTRUCTIONS:** If a small number of SCRs do not have adequate SCM accounting records, rate the response yellow or green as appropriate. If a majority do not have adequate records, rate the response red.

**REFERENCE QUESTIONS:**

- SCM(ID)-015 The contractor's software change control forms are adequate.
- SCM(CC)-002 The procurement activity has implemented adequate software configuration management, based upon regulations, to control the functional and physical characteristics of all CSCIs.
- SCM(CC)-005 The procurement configuration control procedures for the Class I and Class II changes (or equivalent categories) are adequate.
- SCM(SA)-004 The procurement activity configuration status accounting procedures are adequate.
- SCM(SA)-005 The contractor's internal configuration status accounting procedures are adequate.
- SCM(SA)-008 The contractor's automated support tools for configuration status accounting of baselines and internal development identifications are adequate.
- SCM(SA)-009 The contractor's software configuration status accounting forms are adequate.

**4.1.3. QUALITY ASSURANCE**

**FACTOR:** The software quality assurance policies and procedures are effective.

**EXPLANATION:** Assess quality assurance by reviewing the following:

- (1) Sufficient technical quality and integrity guidelines to monitor product quality are established.
- (2) Operational engineering and product impacts of suggested baseline changes are evaluated before they are made.
- (3) Reviews are structured (implemented) to checkpoint the quality of data before they are placed under control.

- (4) A set of planned (implemented) quality inspections are in place to assess software products and process quality.
- (5) Procedures are in place to provide guidance on what is expected in development testing.
- (6) The integrated set of software quality test levels is structured to provide smooth data flow and smooth transition of responsibility.
- (7) Independent observations of software product and process quality are planned (implemented).
- (8) Software management metrics are tracked and used for their intended purpose.

The objective of the software quality organization is to provide a system of checks and balances that identifies potential quality problems before they become significant. The establishment and use of software quality assurance policies and procedures contribute to the contractor's ability to develop operational software. The areas of software quality management include quality reviews, independent quality assurance, independent verification and validation, and test integration of software testing.

**LIFECYCLE APPLICATION:** All phases of the software development effort. During the early stages of software development, software quality assurance standards and procedures should be developed. Planning for IV&V should begin early, starting with an assumption of organic software support and use of the Air Logistics Center (ALC) as the IV&V agent. As development continues, the software quality assurance practices should be monitored to ensure they are applied and to identify shortfalls and improve the policies.

**RELATED METRICS:** SEI assessment results.

**REFERENCE QUESTIONS:**

- SPM(PL)-015 Software quality assessment procedures have been adequately defined to meet management policies and appropriate regulations, conform to standards, and meet performance and quality requirements throughout the system life cycle.
- SPM(PL)-027 Planning for systematic, quantitative, and objectively reportable software tests has been adequate.

**4.1.4. PROGRAM MANAGEMENT**

**FACTOR:** Contractor's program management processes in place to determine implementation methodologies, assess progress, and identify shortfalls are adequate.

**SUBFACTORS:**

RESOURCE FUNCTIONS (4.1.4.1) - The contractor's program methodologies that project, structure, and monitor resources are adequate to complete the software development and developmental testing.

CONTROL FUNCTIONS (4.1.4.2) - The contractor's program management control functions are adequate to ensure software development completion.

**EXPLANATION:** Two indicators which determine contractor's program management process maturity are resource functions and control functions. The management, technical, administrative, quality assurance, and configuration management structures should be tailored and sized to meet the specific needs of the program. The structures should be organized so they are responsive to program needs.

A system of checks and balances exists between program management and configuration management and between program management and quality assurance. Consequently, the configuration

management and quality assurance functions are accomplished even if it slows the development effort. The management structure should, therefore, appropriately allocate authority to the configuration management and quality assurance functions.

**REFERENCE QUESTION:**

       SPM(IM)-005 The implementation methodology used by the contractor's is adequate.

**4.1.4.1. RESOURCE FUNCTIONS**

**FACTOR:** The contractor's program methodologies that project, structure, and monitor resources are adequate to complete the software development and developmental testing.

**SUBFACTORS:**

       **PLANNING (4.1.4.1.1)** - Adequate resources are planned for software development and developmental testing.

       **ORGANIZATION (4.1.4.1.2)** - The software program management organizational structure supports the software development schedule.

       **MONITORING (4.1.4.1.3)** - The software development program management monitoring function is adequate to assess that software development milestones can be met.

**EXPLANATION:** Contractor activity is required to provide certain management information on the project status to the procurement activity. The contractor's software project management support tool environment should be adequate. Review the contractor's use of support tools to perform data collection and processing. The contractor's software resource support methodologies should be adequate to identify shortfalls and make adjustments.

**RESPONSE INSTRUCTIONS:** If no project management tools are being used but the necessary resource management information is being identified, rate the response yellow or green as appropriate. If project management tools and information are not being utilized, rate the response red.

**REFERENCE QUESTIONS:**

- SPM(OS) 008 The contractor's personnel staffing has been adequate throughout the software life cycle phases.
- SPM(OS)-009 The ratio of experienced contractor's project personnel to the total number of project personnel has been adequate throughout the software life cycle phases.
- SPM(OS)-010 The number of contractor's personnel has been adequate through out the software life cycle phases.
- SPM(PL)-004 Computer resources have been adequately addressed as major consider-actions at procurement reviews, audits, and management evaluations.
- SPM(PL)-029 Tracking of computer resource utilization has been adequately planned.
- SPM(IM)-010 The contractor's software project management support tool environment is adequate.

**4.1.4.1.1. PLANNING**

**FACTOR:** Adequate resources are planned for software development and developmental testing.

**EXPLANATION:** Adequate software resource planning and detailed projections-to-complete permit the software manager to minimize software development schedule delays and cost overruns. Development of resource plans utilizing automated tools permits frequent updates. The plans and projec-

tions the software manager develops need to be reviewed for adequacy. A review for adequacy includes checking to ensure the plans will ultimately deliver products that meet user requirements. Comparison of the planning to the current implementation will provide information on the adequacy of resource planning.

The software manager's development plan should reflect the developmental timeline, realistic requirements, and adequate resources to meet the requirements and contain built in contingency pads. The resources development plan should include a monitoring of resources consumed to provide a check on resource effectiveness.

**RESPONSE INSTRUCTIONS:** If resource planning is less than complete, rate the response yellow or green as appropriate. If resource planning was not performed, rate the response red.

**REFERENCE QUESTIONS:**

- SPM(PL)-007 Margins for reserve computer resource capacity to provide for later product improvements are adequate.
- SPM(PL)-029 Tracking of computer resource utilization has been adequately planned.

**4.1.4.1.2. ORGANIZATION**

**FACTOR:** The contractor's software program management organizational structure supports the software development schedule.

**EXPLANATION:** One key to developing and supporting software is to have experienced personnel, especially in the key leadership positions. Experience with the subject system, similar systems, technologically similar problems, the management problems of similar systems, and the interfaces with the subject system results in better managed, higher quality software products. Experienced personnel also shorten the learning curve for less experienced personnel. Therefore, the contractor's personnel experience mix, including management staff, planned for application to the project should be consistent with the productivity realities of the project.

**RESPONSE INSTRUCTIONS:** If the indicators reveal resource and/or schedule deficiencies, rate the response green or yellow as appropriate. If no contractor's resource or schedule management is being performed, rate the response red.

**REFERENCE QUESTION:**

- SPM(OS)-009 The ratio of experienced contractor's project personnel to the total number of project personnel has been adequate throughout the software life cycle phases.

**4.1.4.1.3. MONITORING**

**FACTOR:** The contractor's software development program management monitoring function is adequate to assess the software development milestones can be met.

**EXPLANATION:** Contractor's management monitoring of the software development process and the resources used should be continuous throughout development. The efficient application of resources by management can be achieved only by continual monitoring of resource consumption and progress and milestone achievement status.

The staffing continuity is determined by the rate of turnover of personnel during and across the lifecycle phases. If the same personnel (or at least a reasonable ratio of the same personnel) are not available from phase to phase, then there is likely to be a perturbation in the schedule, cost, func-

tional requirements, and quality of the deliverable products. Turnover of key personnel should be minimal with no sharp variations.

**RESPONSE INSTRUCTIONS:** Interviews with the contractor's software development team and project management are necessary to assess the level at which the contractor management is monitoring the progress. If management is unaware of developmental status to the point of being unaware of missed milestones, rate the response red.

To assess personnel continuity:

Red	50% or more turnover during any one acquisition phase (Concept, Demonstration, Development)
Yellow	20% to 50% turnover during any one acquisition phase
Green	0% to 20% turnover during any one acquisition phase

**RELATED METRICS:** Software Development Personnel.

The software development personnel metric reflects the current status of the development staff to include staff turnover. This indicator compares the following staff counts: total planned, planned experienced, total actual, actual experienced, unplanned losses (total, inexperienced, and experienced). See AFP 800-48 for more information on the calculation and interpretation of the software development personnel metric.

**REFERENCE QUESTIONS:**

- SPM(OS)-008 The contractor's personnel staffing has had continuity throughout the software life cycle phases.
- SPM(PL)-008 Acceptable techniques have been used to estimate and monitor software costs throughout the system life cycle.

**4.1.4.2. CONTROL FUNCTIONS**

**FACTOR:** The contractor's program management control functions are adequate to ensure software development completion.

**SUBFACTORS:**

- CORPORATE POLICIES (4.1.4.2.1)** - The contractors corporate software development policies are adequate to support the software development schedule.
- SUBCONTRACTOR MANAGEMENT (4.1.4.2.2)** - The subcontractor(s) software development effort is adequate.
- COMMUNICATION INTERFACES (4.1.4.2.3)** - The communication interfaces are adequate.

**EXPLANATION:** Contractor's management control procedures should be integrated into the program environment and tailored to the specific needs of the program. Management control functions should be automated to the extent possible. The scope of the management environment should be defined and understood, and there should be a contingency plan to address the completion of tasks on the critical path.

Both software project management and overall program management control is needed to complete the software development. At the management level, control consists of all basic management functions necessary to sustain the development schedule.

The contractor's management structure, management control techniques, and the management environment should be reviewed and, if needed, tailored to the specific program. A chronological review of the developmental effort, first with the SPO software project officer and then with the contractor program and software project managers, will provide the information required to rate this factor.

#### 4.1.4.2.1. CORPORATE POLICIES

**FACTOR:** The contractor's corporate software development policies are adequate to support the software development schedule.

**EXPLANATION:** The contractor's corporate and local policies should agree with program needs, and the contractor's total quality management policy should meet program needs. The contractor's early policy decisions can have major impact on the software development. You should consider several of the following policies that may impact the program: software make or buy policy, developmental hierarchy of testing policy, software or firmware designation policy, and subcontracting of IV&V policy.

Review each of the contractor's corporate policies with the contractor's software project manager. Any policy that could adversely impact OT&E should also be discussed with the program office.

**RELATED METRICS:** SEI Assessment Results.

#### 4.1.4.2.2. SUBCONTRACTOR MANAGEMENT

**FACTOR:** The subcontractor's software development effort is adequate.

**EXPLANATION:** The subcontractor's software development schedule should have the same features as the prime contractor's schedule. In addition, the prime contractor's and subcontractor's schedule, standards, methods, SCM, and quality assurance policies should mesh.

**RESPONSE INSTRUCTIONS:** If there are no subcontractors with software development responsibilities, disregard this factor.

**REFERENCE QUESTIONS:**

- SPM(DM)-015 The contractor's monitor of the subcontractor software design process has been adequate.
- SPM(IM)-007 The contractor's monitor of the subcontractor software implementation process has been adequate.

#### 4.1.4.2.3. COMMUNICATION INTERFACES

**FACTOR:** The communication interfaces are adequate.

**EXPLANATION:** The software development program management function should frequently communicate development status to the software development team, subcontractors, testers, and the program office.

The internal interfaces within the contractor's/subcontractor organization elements should be adequate. Characteristics to be assessed include proper decision process information flow, effectiveness of information flow, and adherence to regulations and guidelines for interface responsibility.

Informal interviews will reveal the extent to which communication is occurring. If communication problems are present, they will be apparent. Identification of the cause is beyond the scope of this SOA.

**RESPONSE INSTRUCTIONS:** Rate the response red if communication problems put software development, DT&E, or OT&E at risk.

**REFERENCE QUESTIONS:**

- SPM(OS)-015 Internal interfaces among contractor's organization elements have been adequate
- SPM(IM)-002 The procurement test organization interface with the contractor's is adequate enough to ensure success of the system integration tests.
- SPM(PI)-012 The IV&V agency external interfaces are adequate.
- SPM(PI)-013 The software configuration management interfaces among all activities' management components for the subject system are adequate.
- SPM(PI)-014 The software quality assurance management interfaces among all activities' management components for the subject system are adequate.
- SCM(AR)-008 The contractor's configuration management tool support facilitates the audit/review of the process by which changes are incorporated into configuration identifications.

#### 4.1.5. TESTING

**FACTOR:** The testing processes are adequate to ensure software is being tested as scheduled, is sufficiently mature, and is satisfying user requirements.

**EXPLANATION:** A mature testing process is one that ensures all the software is tested, the tests will identify any problems, as testing continues the software failure rates decrease, and the user requirements are being met. It should be apparent from the description of the software tests conducted and their results whether or not previous goals have been met and test objectives have been satisfied. Vague references to "successful software results" or "no problems with the software" should not be acceptable. In order to evaluate the progress of software testing to date, there must be explicit reference to a systematic, scientifically sound approach to carrying out the test, the relationship between the systematic test approach and the test objectives for the current phase, the results of the test, and the plans for resolution of errors.

You should review the STP and the STD documents to determine what testing is anticipated. The STP will describe the nature and extent of the testing to be performed. The testing described in these documents should be comprehensive (testing all processing flows, all decision points, and all functions) to the maximum extent allowed by program resources. In addition, the initial testing should be stand alone, in the modules are tested independent from related code. This initial testing should be followed by procedures that "build" to test of grouped modules, until finally an overall integration test of the software is planned. The test planning should indicate tests for both valid and invalid data, as well as stress test and security tests (where applicable).

The STP will have a test schedule you can use to match against current schedules to determine if the testing required by the documents is being accomplished. A review of Software Development Files (if required by the SOW) may also provide an indication of the level and amount of testing being performed on the software.

To produce a valid test, validated test data must be available. Before testing begins, the test data should be reviewed to ensure test completeness, reproducibility, integrity, and predictability.

**REFERENCE QUESTIONS:**

- SPM(PL)-026 The principles and methodologies provided in the regulations have been appropriately incorporated into the software test and evaluation plans.
- SPM(PL)-027 Planning for systematic, quantitative, and objectively reportable software tests has been adequate.
- SPM(DM)-008 The number of software requirements which cannot be tested are minimal.
- SPM(IM)-013 The contractor's application test software tool environment is adequate.
- SPM(TS)-002 The software test process for DT&E has followed the guidelines in the TEMP.
- SPM(TS)-009 The software test approach and methodologies employed are clearly described in the software test documentation and appear to be effective.
- SPM(TS)-010 The software features to be tested and not to be tested are clearly described in the software test documentation.

**4.2. PRODUCTS**

**FACTOR:** Development and maturity of the software products is adequate to ensure properly configured operational software.

**SUBFACTORS:**

DOCUMENTS (4.2.1) - The software documentation set has progressed to the appropriate stage of completion.

SOFTWARE (4.2.2) - The software code is acceptable for the current stage of development.

**EXPLANATION:** To assess the software development products, review the software documentation and source code for problem areas. Assess problem areas to determine their impact.

**4.2.1. DOCUMENTS**

**FACTOR:** The software documentation set has progressed to the appropriate stage of completion.

**SUBFACTORS:**

SDP (4.2.1.1) - The content of the SDP is acceptable for the current stage of development.

SSDD (4.2.1.2) - The content of the SSDD is acceptable for the current stage of development.

INTERFACE DESIGN DOCUMENT (IDD) (4.2.1.3) - The content of the IDD is acceptable for the current stage of development.

SQPP (4.2.1.4) - The content of the SQPP is acceptable for the current stage of development.

SCMP (4.2.1.5) - The content of the SCMP is acceptable for the current stage of development.

SDD (4.2.1.6) - The content of the SDD is acceptable for the current stage of development.

STP (4.2.1.7) - The content of the STP is acceptable for the current stage of development.

SOFTWARE REQUIREMENTS SPECIFICATION (SRS) (4.2.1.8) - The content of the SRS is acceptable for the current stage of development.

INTERFACE REQUIREMENTS SPECIFICATION (IRS) (4.2.1.9) - The content of the IRS is acceptable for the current stage of development.

**EXPLANATION:** The status of the contractor's software documents can be reviewed to provide insights to product maturity. The focus of this portion is on whether the documents are available (when they should be and in the appropriate form (e.g., draft, final)), contain the appropriate computer resources information, and are updated to reflect program changes.

**REFERENCE QUESTIONS:**

SPM(DM) - 016 The design specifications for the software products contain adequate information to implement the software with the required functionality and within the schedule and budget requirements.

SCM(ID) - 017 The documentation which collectively identifies the content of a configuration item is adequate.

**4.2.1.1. SDP**

**FACTOR:** The content of the SDP is acceptable for the current stage of development.

**EXPLANATION:** The SDP describes the contractor's plans for conducting software development. The SDP forms the comparison baseline for most of the contractor documentation. Ensure the following computer resource information is in the SDP:

- (1) An overview of the software project organization structure.
- (2) An identification of the number of personnel necessary to complete the software development effort.
- (3) A description of the organizational responsibilities for performing the software engineering activities.
- (4) A description of each software development activity of the project.
- (5) The impact of any security or safety requirements.
- (6) The proposed corrective action process.
- (7) The software tools necessary to perform the software engineering activities.
- (8) The software product development activities.
- (9) Plans for formal qualification testing.
- (10) A description of the software product evaluation activities.
- (11) A description of the software configuration management activities.

Significant problems with the schedule presented in the SDP could indicate serious development or maturity issues. Ensure the schedule indicates the following:

- (1) Dependencies between tasks.
- (2) The critical path for the software development schedule.
- (3) Schedule milestones as measurable events.
- (4) Cost estimates are updated as the development proceeds.
- (5) Funding is available to support the estimated cost and schedule.
- (6) The cost and schedule status reports are useful management tools (e.g., at the appropriate level of detail, accurate, and up to date).
- (7) Time for software development activities.
- (8) Adequate time to address action items from reviews (i.e., PDR).
- (9) Adequate slack time to accommodate corrections of any deficiencies discovered during testing (e.g., unit test, integration test, DT&E).

To determine all activities are included in the schedule, review the WBS to identify each task. The WBS should include software development activities such as major reviews and milestones and be used as the basis for schedule and cost estimates. The WBS should be updated throughout the program to reflect significant changes.

#### 4.2.1.2. SSDD

**FACTOR:** The content of the SSDD is acceptable for the current stage of development.

**EXPLANATION:** The SSDD is the primary document for the system design. It should contain the following computer resources information:

- The description of each CSCI includes a summary of the CSCI purpose, a definition of the interfaces external to the CSCI, and a matrix of the system requirements allocated to the CSCI.
- The software related interfaces associated with each CSCI include: interfaces external to each CSCI, CSCI to CSCI interfaces, and CSCI to HWCI interfaces.
- The description of the processing resources specifies the hardware, programming, design, coding, and utilization characteristics of each resource. For example, the processing capacity (absolute and spare), growth capacities, and diagnostic capabilities.
- The requirements traceability matrix provides traceability of the system requirements allocated to the CSCIs back to the requirements of the system specification.

#### REFERENCE QUESTIONS:

See AFOTECPAM 99-102, volume 3.

#### 4.2.1.3. IDD

**FACTOR:** The content of the IDD is acceptable for the current stage of development.

**EXPLANATION:** The IDD specifies the detailed design for one or more interfaces between CSCIs and other configuration items or critical items and is predicated upon the interface requirements specification (IRS). It should (1) contain interface diagrams and a complete data element definition table for each data element to be transmitted across the interface, (2) specify the relative priority of each interface and of each message to be transmitted across each interface, and (3) describe the technical details of each communications protocol associated with each interface.

The data element definition tables for each interface include the following information (as applicable): data element identifier, data element description, CSCI that is the data element source, CSCI that is the data element user, data element unit of measure, valid range of values, accuracy requirements, calculation (or refresh) frequency, validity checks, data type, data format, and data priority.

The technical details of each communication protocol include (as applicable): message fragmentation and reassembly, message formatting, error control and recovery procedures, synchronization, flow control, data transfer rate, routing, transmission services, status, and security.

#### 4.2.1.4. SQPP

**FACTOR:** The content of the SQPP is acceptable for the current stage of development.

**EXPLANATION:** The SQPP documents the contractor's software quality program. The objective of the software quality program is to improve the quality of the deliverable products, the nondeliverable products, and the software development processes. The software quality program includes planning, assessment, reporting, and follow-up activities. The SQPP should describe the software quality environment, organizational structure, procedures and tools, and program activities.

The SQPP describes the products and processes (and tools) of the software quality program activities. The software quality program activities include assessments of the following: documentation, storage and shipping of deliverable software, software development libraries, software development files, software change requests, software testing, software installation and checkout, nondeliverable software, and subcontractor developed software.

#### 4.2.1.5. SCMP

FACTOR: The content of the SCMP is acceptable for the current stage of development.

EXPLANATION: The SCMP identifies the responsibilities of the software configuration management (SCM) organization and describes the policies and procedures of that organization. It should define the allocation of CM resources, specify the planned (or existing) CM policies and procedures, identify the planned (or existing) CM tools, and establish the function and procedures associated with the CCB.

The procedures for controlling software problems and changes should be addressed in the SCMP. These procedures should include a description of the software configuration control flow and audit mechanisms, a detailed description of the software problem change report, and the responsibilities and procedures of the software change authority, the CCB.

#### 4.2.1.6. SDD

FACTOR: The content of the SDD is acceptable for the current stage of development.

EXPLANATION: A separate SDD is written for each CSCI. Each SDD should describe the internal organization of the CSCI, the CSUs of each CSC, and the global data elements of the CSCI.

The description of the internal organization includes summarization of the CSCs and sublevel CSCs, the relationships between CSCs, memory allocation and processing time, definition of the system requirements allocated to each CSC, description of the preliminary design in terms of execution control and data flow, derived requirements, and design constraints.

The description of the CSUs of each CSC includes summarization of the CSUs, relationship between CSUs, and statement of the design requirements for each CSU. Examples of CSU design requirements include I/O data elements, algorithms, and error handling.

The SDD should be consistent with the information in the SRS.

#### REFERENCE QUESTIONS:

From AFOTECPAM 99-102, volume 3.

- D-20 The documentation adequately describes the external interfaces.
- D-29 Any dynamic allocation of resources is explained in the documentation.
- D-32 Storage requirements for each major function of the program are adequately described in the documentation.
- D-59 It is easy to trace the program control flow at all system levels.

#### 4.2.1.7. STP

FACTOR: The content of the STP is acceptable for the current stage of development.

EXPLANATION: The STP describes the formal qualification test plans for one or more CSCIs. The STP also identifies the software test environment resources required for formal qualification testing.

It should describe the plan for implementing and controlling the resources necessary to perform formal qualification testing, describe the contractor's plans for controlling and maintaining each item in the test environment, describe the formal qualification test for each CSCI, contain (or reference) the formal qualification test schedule, and describe the data reduction and analysis procedures.

Software test environment resources include software items needed to perform formal qualification testing such as operating systems, compilers, code auditors, dynamic path analyzers, test drivers, preprocessors, test data generators, and post processors.

The STP contains contractor plans for the test environment to include installing and testing each item prior to its use.

The formal qualification test requirements for each CSCI should address the following points:

- (1) General test requirements (e.g., error detection, error recovery, boundary conditions).
- (2) Test classes (e.g., stress, timing, erroneous input, maximum capacity tests).
- (3) Test levels (e.g., CSC-CSC, CSCI-CSCI, CSCI-HWCI, system level).
- (4) Test definitions (e.g., qualification method, requirements cross-references, assumptions).
- (5) Test schedule.

#### REFERENCE QUESTIONS:

From AFOTECPAM 99-102, volume 3 and volume 2.

D-48 The program test plan is adequately described in the documentation.

D-49 A useful set of test procedures for high levels of program testing is contained in the documentation.

D-50 A useful set of test procedures for low levels of program testing is contained in the documentation.

D-51 The limitations and incompleteness of the test procedures are described in the documentation.

D-52 The sample test data are adequately described in the documentation.

D-53 Program support tools that aid in testing the program are adequately documented.

D-54 The documentation describes software test probes that aid in identifying processing performance.

SPM(TS)-011 The traceability software features tested/not tested to the software functional requirements is described in the software test documentation.

SPM(TS)-013 The software test criteria used to determine whether each test has passed or failed are clearly specified in the software test documentation.

SPM(TS)-014 The personnel groups responsible for the software tests are adequately identified in the software test documentation.

SPM(TS)-015 The high risk assumptions of the software testing approach along with contingency plans for each such assumption is adequately described in the software test documentation.

SPM(TS)-016 The schedule for software test milestones is adequately specified in the software test documentation.

SPM(TS)-017 Software testing is adequately prioritized in the software test approach according to mission criticality concerns.

SPM(TS)-018 The software test environment is adequately identified in the software test documentation and is adequate for accomplishing the required testing.

#### 4.2.1.8. SRS

**FACTOR:** The content of the SRS is acceptable for the current stage of development.

**EXPLANATION:** Ensure the SRS specifies:

- The engineering and qualification requirements for a CSCI.
- The engineering requirements necessary to ensure proper development of the CSCI, external interfaces of the CSCI
- All of the capability requirements that the CSCI must satisfy
- Interfaces between the capabilities identified in the previous step
- CSCI data element requirements
- Requirements for adapting the CSCI to site-unique conditions and to changes in the system environment
- Sizing and timing requirements
- Safety and security requirements
- The applicable human factors engineering requirements for the CSCI.

#### 4.2.1.9. IRS

**FACTOR:** The content of the IRS is acceptable for the current stage of development.

**EXPLANATION:** The IRS specifies the requirements for one or more interfaces between a particular CSCI and other configuration items or critical items in the system of which it is a part. An IRS is used to provide interface requirements in a separate document in the following circumstances: (1) there are many interfaces, (2) there are many development groups involved in communicating the requirements, (3) the complexity of the interface is such that a separate IRS will simplify communications, or (4) two or more contractors are developing the requirements. When an IRS is used to specify requirements for interfaces, the detailed design of the interfaces is provided in an IDD.

Interface block diagrams depict the relationship of the CSCI to the other HWCI's, CSCIs, or critical items in the system for which interfaces are specified.

The IRS specifies the methods, techniques, tools, facilities, and acceptance tolerance limits necessary to establish that each interface satisfies the identified requirements.

#### 4.2.2. SOFTWARE

**FACTOR:** The software code is acceptable for the current stage of development.

**SUBFACTOR:**

RELIABILITY (4.2.2.1) - The rate of software failures will not impact the start or completion of OT&E.

SCHEDULES (4.2.2.2) - The software is being developed according to schedule.

SPARE CAPACITY (4.2.2.3) - Margins for reserve computer resource capacity are adequate.

MAINTAINABILITY (4.2.2.4) - The software and associated documentation are easily modified.

**EXPLANATION:** The status of the software can be reviewed to provide insight as to the maturity of the products. To be mature, the software should show a decreasing failure rate trend, be on schedule in terms of development, should be reaching closure on user spare capacity requirements, and should be sufficiently maintainable.

**RESPONSE INSTRUCTIONS:** If any of the above are unacceptable and there is no plan, you should answer red. If an adequate plan exists, answer yellow.

#### 4.2.2.1. RELIABILITY

**FACTOR:** The rate of software failures will not impact the start or completion of OT&E.

**EXPLANATION:** Reliability is one of the measures of software maturity and an indication of whether the software development effort will support the start of OT&E. Reliability is defined as the extent to which the software will perform its intended function without failure within a specified time period. In the software failure rate is decreasing, the reliability will be increasing, and the software can be interpreted as becoming more mature.

For most applications, acceptable levels range from one to three errors per thousand lines of code. Unfortunately, an OA may be performed long before delivered lines of code are known. For the lack of any other information, the evaluator may be able to obtain information that tracks the error rate for developed (at the time of the OA) lines of code and, hence, determine the trend of the rate.

**RELATED METRICS:** Software reliability metrics are defined in AFOTEC-PAM 99-102, volume 6.

#### 4.2.2.2. SCHEDULES

**FACTOR:** The software is being developed according to schedule.

**EXPLANATION:** Examine the schedules for source code development and testing and ensure the code is being developed on schedule, or if it is behind schedule, there are contingency plans to handle any code development problems within the schedule. If a software product's development is not meeting its development schedule, this can be an indicator of immature software. Reasons for not meeting the schedule include: more errors are being found and corrected than were anticipated or required access to computer hardware was not being met.

**RESPONSE INSTRUCTIONS:** You must decide whether lack of software progress toward meeting its schedule is a result of an inadequate schedule or the schedule is actually correct but other issues/problems have surfaced related to progress toward maturity. For example, poor configuration control, lack of adequate software standards and practices, assignment of unskilled software personnel, and many other factors could cause the introduction of more errors into developed software that would have been anticipated when the schedule was built. Your best approach is to determine the conditions or assumptions under which the schedule was made and to make a judgment about the cause of the problem. It is expected problems not related to current schedule adequacy would surface under other maturity factors. However, if this is not the case, such problems could be reported here. If software is not meeting its development schedule, rate this factor yellow or red, as appropriate.

#### 4.2.2.3. SPARE CAPACITY

**FACTOR:** Margins for reserve computer resource capacity are adequate.

**EXPLANATION:** Requirements for computer margins and the initial values for timing and sizing are established by the using command's ORD. These margins then evolve throughout the software development effort. Margins should be established for memory, external storage, task utilization, terminal usage, performance parameters, and so forth. The margin of reserve is very important for software supportability, since changes will usually require consumption of some of the reserve.

**RELATED METRICS:** Computer Resource Utilization. The computer resource utilization metric tracks resource usage and reserve capacity available for each computer resource. For each computer resource, three subindicators should be tracked. These three subindicators are central processing unit (CPU) utilization (planned vs. actual percentage of CPU execution time consumed), memory utilization (planned vs. actual percentage of memory consumed), and I/O utilization (planned vs.

actual percentage of I/O resources consumed). See AFP 800-48 for more information on the calculation and interpretation of the computer resource utilization metric.

#### 4.2.2.4. MAINTAINABILITY

**FACTOR:** The software and associated documentation are easily modified.

**EXPLANATION:** Maintainability is the effort needed to change software, where change could occur to correct errors, add system capabilities, delete features, or modify the software to become compatible with hardware changes. It is a prerequisite to start and complete OT&E. Maintainable software results from the use of a sound development process to build a quality product.

If products are available, perform some type of AFOTECPAM 99-102, volume 3 evaluation. If products are unavailable, look at the contractor's coding standards and evaluate whether the standards will produce products that will score well on a volume 3 evaluation.

**RELATED METRICS:** There are several software metrics that provide an indication of the maintainability of the software product. Simple metrics include the effort expended to correct errors or the amount of time problem reports are remaining open. The results of an AFOTECPAM 99-102, volume 3 evaluation provide metric values for the maintainability of the documentation, source code, and implementation. Complexity metrics have been established in Automated Software Tools System (ASSETS) and can be used to aid in evaluating this factor.

### 5. REQUIREMENTS TRACEABILITY

The one significant computer resource issue not addressed in other categories is requirements traceability. It should be noted that "computer resource requirements," as used in this section, includes not only specific computer resource requirements, but also requirements (or functions) allocated to software.

**FACTOR:** There are no missing computer resource requirements that would adversely impact the system.

#### **SUBFACTORS:**

ORD/RCM TO SYSTEM SPECIFICATION TRACEABILITY (5.1) - The user's computer resources requirements are traceable from the ORD/RCM to the system specification.

DEVELOPMENT CONTRACTOR TRACEABILITY (5.2) - The contractor has adequately traced the user's computer resources requirements from the system specification through the development documents.

#### 5.1. ORD/RCM TO SYSTEM SPECIFICATION TRACEABILITY

**FACTOR:** The user's computer resources requirements are traceable from the ORD/RCM to the system specification.

**EXPLANATION:** The ORD/RCM should identify computer resources requirements. It should address all mission critical and support computer resources, including automated test equipment. The capabilities desired for integrated computer resources support should be described. It should describe any unique interface requirements, documentation needs, and special software certifications. Each of the requirements in the ORD/RCM should be contained in the system specification. Methods are available that document relationships between requirements, and create a requirements traceability matrix to assist the user in verifying requirements allocation. This process can be done manually or by using automated tools, but a disciplined method is needed. The output of the

method should provide some form of a requirements traceability matrix. You should use this matrix to determine each computer resources requirement in the ORD/RCM is contained in the system specification.

In addition to the specific computer resources requirements, many of the user's operational effectiveness and operational suitability performance requirements identified in the ORD/RCM may be allocable, in whole or in part, to software. The system specification must contain each of these ORD/RCM requirements.

## 5.2 DEVELOPMENT CONTRACTOR TRACEABILITY

**FACTOR:** The contractor has adequately traced the user's computer resources requirements from the system specification through the development documents.

**EXPLANATION:** Requirements in the system specification are allocated to lower tier specifications and development documents. Each requirement in the system specification must be traceable through all the other development documents. Methods are available that document relationships between requirements and create a requirements traceability matrix to assist the user in verifying requirements allocation. This process can be done manually or by using automated tools, but a disciplined method is an absolute necessity for complex systems. You should determine the contractor has a disciplined method.

The computer resources requirements should be traceable from the system specification to the system design document, from the system design document to a software requirements specification, and then from the software requirements specification to the software design document. You should select a representative sample (as large as feasible) of computer resources requirements in the system specification and trace each one through the hierarchy of development documents using the contractor's requirements traceability method.

### REFERENCE QUESTIONS:

       SPM(DM)-007 The number of software requirements which cannot be traced to an end item product is minimal.

(From AFOTECPAM 99-102, volume 3)

- D-55 System requirements are easily traceable to the actual function/CSCI that implements the requirement.
- D-56 A function's (or CSCI's/CSC's) description is easily traceable to the detailed descriptions of the modules (or CSUs) performing that function.